



Entity 설계 & ERD

주요 엔티티 정의

employees (직원)

속성명	한글명	설명	제약조건
id	식별자	고유 ID	필수, 고유, 자동 생성
employee_number	직원 번호	사원 번호	필수, 고유, 자동 부여, 수정 불가
employee_name	직원 이름	직원 이름	필수
email	이메일	직원 이메일	필수, 중복 금지
department_id	부서 ID	departments 테이블의 PK	필수, FK
position	직함	직원 직함	
hire_date	고용 날짜	입사일	필수
employee_state	직원 상태	직원의 재직 상태	기본값 ACTIVE
profile_image_id	프로필 이미지 ID	file_metadata 테이블의 PK	FK

departments (부서)

속성명	한글명	설명	제약조건
id	식별자	고유 ID	필수, 고유, 자동 생성
department_name	이름	부서 이름	필수, 고유
department_description	설명	부서 설명	필수
department_established_date	설립일	부서 설립일	필수

change_logs (직원 정보 수정 이력)

속성명	한글명	설명	제약조건
id	식별자	고유 ID	필수, 고유, 자동 생성
employee_id	대상 직원 ID	변경 이력이 발생한 직원의 ID (employees 테이블의 id와 외래키 연결)	필수, 외래키, 참조: employees(id)
type	이력 유형	이력 변경 유형 (CREATED, UPDATED, DELETED)	필수, Enum
memo	메모	사용자가 입력한 메모 내용	VARCHAR(255)
ip_address	IP 주소	요청을 발생시킨 클라이언트의 IP 주소	필수, IPv4 주소 저장, VARCHAR(20)
created_at	이력 생성 일시	변경 이력이 생성된 시간	필수, 자동 생성
details	변경 상세 내용	변경된 속성명, 이전 값, 이후 값 목록 ex: [{"propertyName": "직함", "before": "사원", "after": "대리"}]	JSONB 형식

change_log_details (직원 정보 수정 이력 상세 내용)

속성명	한글명	설명	제약조건
id	식별자	고유 ID	필수, 고유, 자동 생성
change_log_id	변경 이력 ID	변경 이력의 ID (change_logs 테이블의 id와 외래키 연결)	필수, 외래키, 참조: change_logs(id)
property_name	변경 항목	변경된 속성	필수, VARCHAR(50)
before_value	변경 전 값	변경 전 속성 값	VARCHAR(255)
after_value	변경 후 값	변경 후 속성 값	VARCHAR(255)

file_metadata (파일 메타정보)

속성명	한글명	설명	제약조건
id (PK)	식별자	고유 ID	필수, 고유 자동 생성
file_name	파일 이름	이미지 파일 이름	필수

속성명	한글명	설명	제약조건
content_type	파일 타입	이미지 MIME 타입 (jpg, jpeg, png, gif), csv, .log타입	필수, 이미지 타입, csv, .log파일만 허용
file_size	파일 크기	이미지 파일 크기	필수

backup_histories (백업 이력)

속성명	한글명	설명	제약조건
id (PK)	식별자	고유 ID	필수, 고유 자동 생성
operator	작업자	백업 작업자의 IP	필수
started_at	시작 시간	백업 시작 시간	필수
ended_at	종료 시간	백업 종료 시간	-
status	상태	백업 상태 (진행중, 완료, 실패, 건너뛸)	필수
backup_file_id (FK)	백업 파일	백업 파일의 CSV 파일 혹은 에러 로그의 .log 파일 메타 정보의 참조 ID	외래키

employee_stats (일별 직원수 집계 / 전체 추이용)

속성명	한글명	설명	제약조건
id	식별자	고유 ID	필수, 고유 자동 생성
stat_date	해당 일자	집계를 몇 일에 했는지	필수, 고유
employee_count	직원 수	집계 일 기준 직원 수	필수
created_at	생성시간	집계 데이터 생성 시간	필수
joined_employee_count	해당 일 입사자 수	해당 일에 몇 명이 입사했는지	필수
left_employee_count	해당 일 퇴사자 수	해당 일에 몇 명이 퇴사했는지	필수

department_stats (부서별 직원 분포 통계)

속성명	한글명	설명	제약조건
id	식별자	고유 ID	필수, 고유 자동 생성

속성명	한글명	설명	제약조건
stat_date	해당 일자	집계를 며칠에 했는지	필수, 복합 Unique(stat_date, department_id, employee_state)
employee_count	직원 수	집계 일 기준 부서 인원 수	필수
created_at	생성시간	집계 데이터 생성 시간	필수
joined_employee_count	해당 일자 부서 입사자 수	해당 일에 몇 명이 부서에 입사했는지	필수
left_employee_count	해당 일자 부서 퇴사자 수	해당 일에 해당 부서에서 몇 명이 퇴사했는지	필수
department_name	부서 이름	통계 대상 부서의 이름	필수, 복합 Unique(stat_date, department_name, employee_state)
employee_state	직원 상태	직원 상태 별로 서로 다른 통계를 내기 위함	필수, 복합 Unique(stat_date, department_name, employee_state)

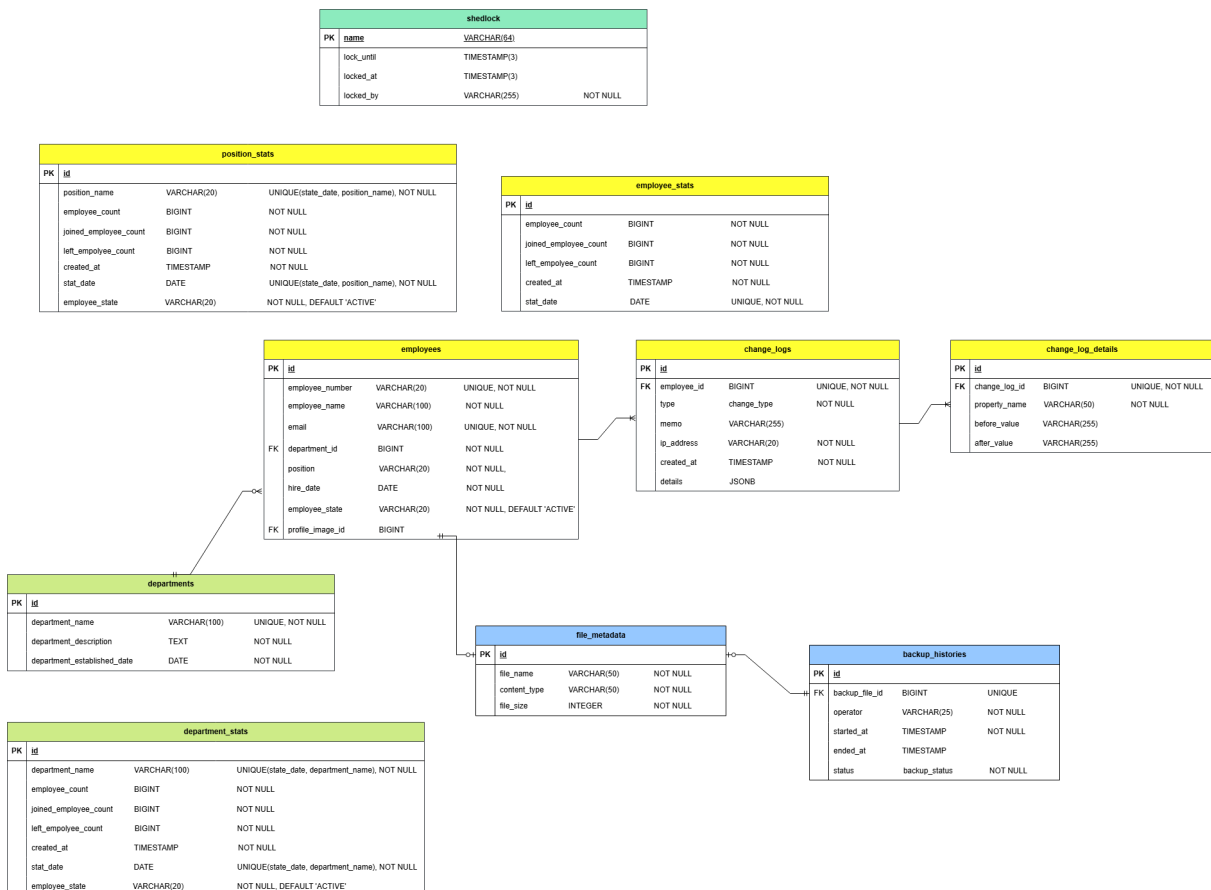
position_stats (직무별 직원 분포 통계)

속성명	한글명	설명	제약조건
id	식별자	고유 ID	필수, 고유 자동 생성
stat_date	해당 일자	집계를 며칠에 했는지	필수, 복합 Unique(stat_date, position_name, employee_state)
employee_count	직원 수	집계 일 기준 직무 인원 수	필수
created_at	생성시간	집계 데이터 생성 시간	필수
joined_employee_count	해당 일자 직무 입사자 수	해당 일에 몇 명이 해당 직무로 입사했는지	필수
left_employee_count	해당 일자 직무 퇴사자 수	해당 일에 해당 직무 인원이 얼마나 퇴사했는지	필수

속성명	한글명	설명	제약조건
position_name	직무 이름	통계 대상 직무의 이름	필수, 복합 Unique(stat_date, position_name, employee_state)
employee_state	직원 상태	직원 상태 별로 서로 다른 통계를 내기 위함	필수, 복합 Unique(stat_date, position_name, employee_state)

ERD

[https://app.diagrams.net/#G1PA5M26DpZmc0RK-sbRALYZGzpEWHcH8y#{_pageId"%3A"uN84papr1lelol6XfhSQ"}](https://app.diagrams.net/#G1PA5M26DpZmc0RK-sbRALYZGzpEWHcH8y#{_pageId)



DDL 스크립트

employees

```
CREATE TABLE employees (  
  id BIGSERIAL PRIMARY KEY,  
  employee_number VARCHAR(20) UNIQUE NOT NULL,  
  employee_name VARCHAR(100) NOT NULL,  
  email VARCHAR(100) UNIQUE NOT NULL,  
  department_id INTEGER NOT NULL,  
  position VARCHAR(20) NOT NULL,  
  hire_date DATE NOT NULL,  
  employee_state VARCHAR(20) NOT NULL DEFAULT 'ACTIVE',  
  profile_image_id INTEGER,  
  
  CONSTRAINT fk_department  
    FOREIGN KEY (department_id)  
    REFERENCES departments(id),  
  
  CONSTRAINT fk_profile_image  
    FOREIGN KEY (profile_image_id)  
    REFERENCES file_metadata(id)  
);
```

departments

```
CREATE TABLE departments (  
  id BIGSERIAL PRIMARY KEY,  
  department_name VARCHAR(100) NOT NULL UNIQUE,  
  department_description TEXT NOT NULL,  
  department_established_date DATE NOT NULL  
);
```

change_logs

```

CREATE TYPE change_type AS ENUM ('CREATED', 'UPDATED', 'DELETED');

CREATE TABLE change_logs (
  id BIGSERIAL PRIMARY KEY,
  employee_id BIGINT,
  type change_type NOT NULL,
  memo VARCHAR(255),
  ip_address VARCHAR(20) NOT NULL,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,

  CONSTRAINT fk_change_logs
    FOREIGN KEY (employee_id)
    REFERENCES employees(id)
    ON DELETE SET NULL
);

```

change_log_details

```

CREATE TABLE change_log_details (
  id BIGSERIAL PRIMARY KEY,
  change_log_id BIGINT NOT NULL,
  property_name VARCHAR(50) NOT NULL,
  before_value VARCHAR(255),
  after_value VARCHAR(255),

  CONSTRAINT fk_change_log_details
    FOREIGN KEY (change_log_id)
    REFERENCES change_logs(id)
    ON DELETE CASCADE
);

```

file_metadata

```

CREATE TABLE file_metadata(
  id BIGSERIAL PRIMARY KEY,
  file_name VARCHAR(50) NOT NULL,

```

```
content_type VARCHAR(50) NOT NULL,  
file_size INTEGER NOT NULL  
);
```

backup_histories

```
CREATE TYPE backup_status AS ENUM ('IN_PROGRESS', 'COMPLETED', 'FAIL'  
  
CREATE TABLE backup_histories(  
  id BIGSERIAL PRIMARY KEY,  
  operator VARCHAR(25) NOT NULL,  
  started_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,  
  ended_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  status backup_status NOT NULL,  
  backup_file_id BIGINT,  
  
  CONSTRAINT backup_file_fk  
    FOREIGN KEY (backup_file_id)  
    REFERENCES file_metadata(id)  
    ON DELETE SET NULL  
);
```

employee_stats

```
CREATE TABLE employee_stats (  
  id BIGSERIAL PRIMARY KEY,  
  employee_count BIGINT NOT NULL,  
  joined_employee_count BIGINT NOT NULL,  
  left_employee_count BIGINT NOT NULL,  
  created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  stat_date DATE NOT NULL UNIQUE  
);
```

department_stats


```

CREATE TABLE department_stats (
  id BIGSERIAL PRIMARY KEY,
  department_name VARCHAR(100) NOT NULL,
  employee_count BIGINT NOT NULL,
  joined_employee_count BIGINT NOT NULL,
  left_employee_count BIGINT NOT NULL,
  created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  stat_date DATE NOT NULL,
  employee_state VARCHAR(20) NOT NULL DEFAULT 'ACTIVE',

  CONSTRAINT uq_department_state_date
    UNIQUE (stat_date, employee_state, department_name)
);

```

position_stats

```

CREATE TABLE position_stats (
  id BIGSERIAL PRIMARY KEY,
  position_name VARCHAR(20) NOT NULL,
  employee_count BIGINT NOT NULL,
  joined_employee_count BIGINT NOT NULL,
  left_employee_count BIGINT NOT NULL,
  created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  stat_date DATE NOT NULL,
  employee_state VARCHAR(20) NOT NULL DEFAULT 'ACTIVE',

  CONSTRAINT uq_position_state_date
    UNIQUE (stat_date, employee_state, position_name)
);

```

shedlock

```

CREATE TABLE shedlock (
  name VARCHAR(64) PRIMARY KEY,
  lock_until TIMESTAMP(3) NULL,
  locked_at TIMESTAMP(3) NULL,

```

```
locked_by VARCHAR(255) NOT NULL  
);
```